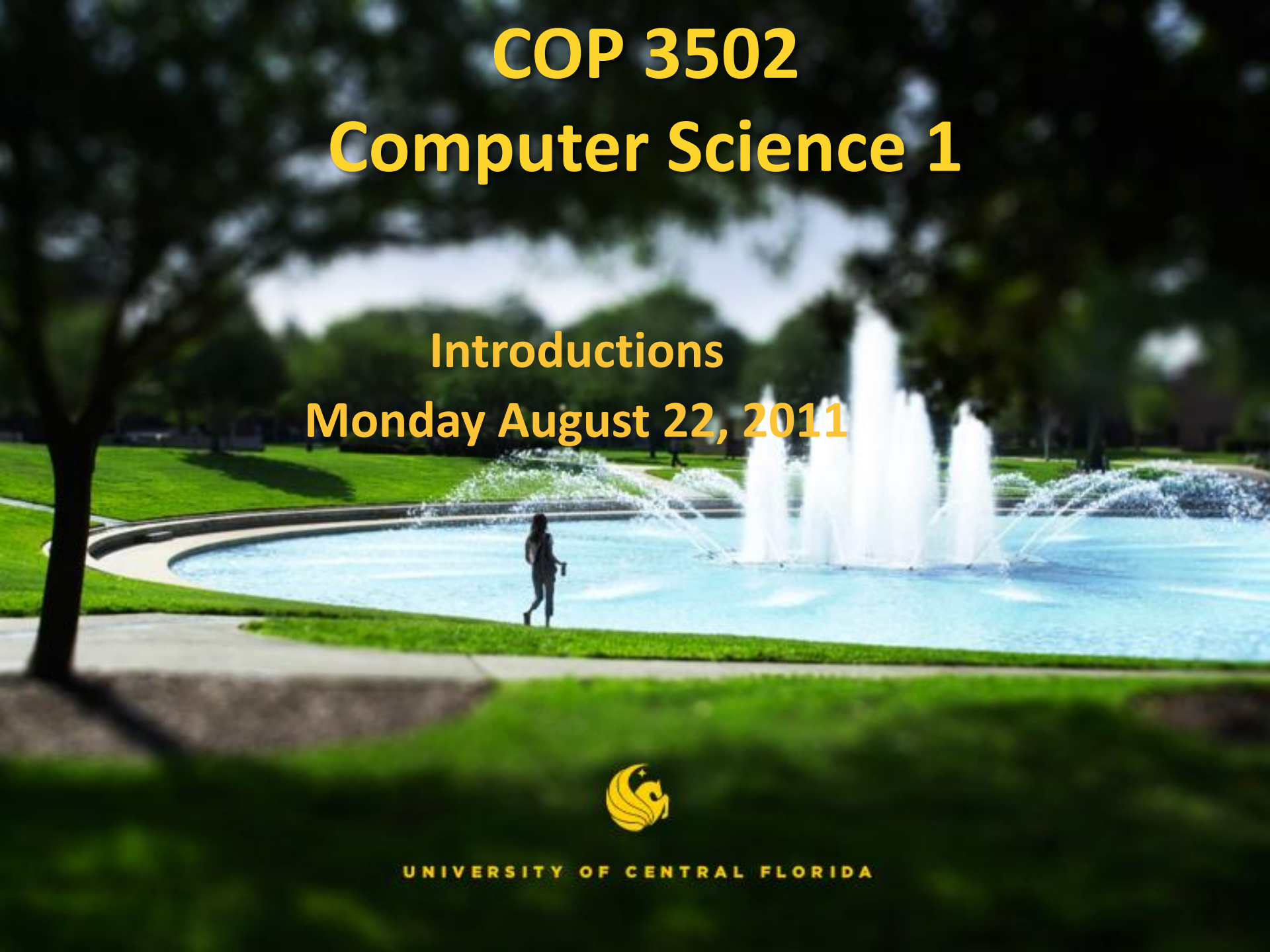

COP 3502
Computer Science 1
Introductions
Monday August 22, 2011
UNIVERSITY OF CENTRAL FLORIDA

Introduction

- Welcome to Computer Science 1!
- Instructor: Sarah Buchanan

Announcements

- No Lab this week, but there will be Labs next Monday!
- I will post Assignment #1 later today.

Website vs Webcourses

- The website: www.cs.ucf.edu/courses/cop3502/fall2011/
  - Contains all of the course content: lectures, assignments etc.

- Webcourses is where all of your assignments are submitted, and grades are stored.

Contact Info

- If you have any assignment related questions, please ask the TA's.
- If you need to contact me the best way is email: sbuchanan@knights.ucf.edu
- OR office hours:
  - Monday and Wednesday 10:30am – 12pm
  - (Right after class)

What is this class about?

- COP 3223 is a prereq for this class
  - It's expected that you know C language basics
    - Including variable declarations, conditional expressions, if statements, loops, functions, and arrays.
  - We will review:
    - Pointers, 2D arrays, structures, and linked lists.

What is this class about?

- Goals of this class:
  1. Improving knowledge of standard data structures and abstract data types.
  2. Improving knowledge of standard algorithms used to solve several classical problems.
  3. Covering some of the Mathematics that is useful for the analysis of algorithms.
  4. Analyzing the efficiency of solutions to problems.

What is this class about?

- This class will NOT be used to teach you C.
  - You will need to freshen up on your C very quickly.
  - If you want extra help on C, you can buy a C – language book or find online resources.
  - “C by Dissection” is a good reference book.

What is this class about?

- So, whereas in COP 3223 we only cared if we found a solution to a problem,
  - In ***COP 3502*** we will also learn ***standard ways to solve problems*** and also ***how to analyze the efficiency of those solutions***.
- Additionally, we will expand upon our knowledge of the C programming language.

Expectations

- Read the C Review posted online.
- Get started on the 1st assignment quickly, it will serve as a quick refresher on C, as it will use reading from a file, structs, enumerations and dynamically allocated arrays.
  - It is due the Wednesday after Labor Day 9/7

Syllabus

- Make sure you read the syllabus carefully so that you know what to expect for the class.

- It's on the course website:
  - http://www.cs.ucf.edu/courses/cop3502/fall2011/

Attendance

- Attendance for class is not mandatory
  - But you will find that if you attend you will do better in the class.
  - To reward students that come to class, I will randomly take attendance up to 8 times, and for each time you are here you can earn a maximum of 5 points on the final exam.
- Attendance for the labs will be taken.
  - There are 12 labs, and only 10 will be counted, so you have 2 freebees (for emergencies such as being sick)

Grades

Item	Percentage
Exam #1	20
Exam #2	20
Final Exam	20
Homework Assignments	25
Lab Participation	15

- So each Exam is 20% of your grade.
- There are 5 assignments, worth 5% each.
- And Lab Participation is 15% of your grade.

Lab Participation

- Each time you attend lab and *participate*
  - you earn participation points towards 10% of your grade.
- As a way for you to measure your understanding of the material, there will be 7 pop quizzes given during the labs throughout the semester.
  - These quizzes will count as 5% of your grade total, so 1% for each quiz, and you can throw out your worst 2 quizzes.

Research Study

- The purpose of this study is to evaluate different methods of teaching data structures to computer science students.
- During 7 pre-determined labs you will be given a lecture using different methods.
  - Before the lab you will be given a pre-test and after the lab you will be given a quiz.
  - The quiz will count towards your grade, but will also contain an evaluation of the lecture that will not be counted towards your grade.
- So we will do our normal lab activity and then at the end of lab you will have a “pop” quiz.
  - You won’t have to study for the pop quizzes, since they will be directly related to the problems and topics covered in lab.

Schedule

- www.cs.ucf.edu/courses/cop3502/fall2011

Cheating

- Cheating will NOT be tolerated
  - Any copying of assignments from each other or on the internet is not allowed.
    - Both people involved are cheating, no matter if you shared your work or received work.
    - Cheating can be as little as 3 lines of code.
  - If you have questions, ask the TA's
    - You can discuss how to do problems with your friends, but don't give code away. Getting code from someone isn't going to help you on how to figure out the problem.
  - You will receive a -25% grade on any assignment or exam that you are caught cheating on.

Example Problem

- We will now go over 2 solutions to a problem:
  - The first is a straightforward solution that a COP 3223 student should be able to come up with
    - Doesn't care about efficiency
  - The second solution is one that a COP 3502 student should be able to come up with after some thought
    - Cares about efficiency
  - This example should illustrate part of the goals of this course.

Example Problem

- Find Max Number of 1's:
  - You're given an n x n integer array where each row is filled with several 1s followed by all 0's.
  - The goal: determine the max number of 1s in any row.

1	1	0	0	0	0
0	0	0	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	1	1	0
1	1	1	1	0	0

Example Problem

- First let's go through a straight-forward solution without using any code:
  - For each row do the following:
    - Start from the beginning of the row, scanning from left to right, until the first zero is encountered, keeping track of how many ones have been seen on that row as you go.
    - If the number of 1's is greater than the previous maximum seen, then update the maximum number of 1s seen.

Straight Forward Solution

maxNumOnes = 5

1	1	0	0	0	0
0	0	0	0	0	0
1	1	1	0	0	0
1	1	1	1	1	0
1	1	0	0	0	0

Example Problem

- Of course this will work… now let's see how long this algorithm will take.
  - Basically, we iterate through each square that contains a 1 in it, as well as the first 0 in each row.
    - If all cells were 0s, we would "visit" n squares total.
    - However, if all the cells were 1s, we would "visit" n^2 squares total.
    - Thus, in the worst case, the number of simple steps this algorithm would take would be approximately n^2, this makes the running time of this algorithm $O(n^2)$.
      - (The meaning of this Big-Oh will be discussed later this semester.)

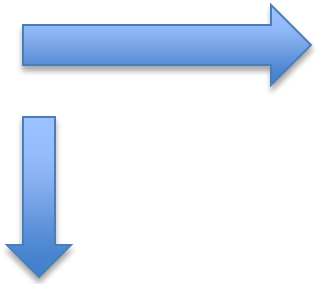
Example Problem

- There seems to be some potential extra work done here.
  - Once we know that one row has 12 1's for example, it seems pointless to start at the beginning of the next row.
  - Why not just start in column 12? If it's a 0, then certainly that row can't be a winner, and if it is a 1, clearly there is no point now to go back and check the previous 11 squares.

Example Problem

- So here is our more efficient COP 3502 style algorithm:

1) Initialize the current row and current column to 0
2) While the current row is less than n
   a) While the cell at the current position is a 1, move to the right (increment the column)
   b) Else, move down (increment the current row)
3) The current column index represents the MAX # of 1's seen.



Example Problem

row/ col	0	1	2	3	4
0	1	1	0	0	0
1	1	1	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0
4	1	1	1	1	0

Example Problem

Row/ col	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	0	0	0	0	0	0
5	1	1	1	1	1	0	0	0	0	0
6	1	1	1	1	1	1	1	0	0	0
7	1	1	1	1	1	1	1	1	0	0
8	1	1	1	1	1	1	1	1	1	1

Example Problem Analysis

- Improved Solution Analysis:
  - How many “steps” will this algorithm take, in terms of n?
    - We can take a maximum number of n-1 steps to the right,
    - and a maximum of n-1 steps down
    - = approximately 2n steps total
    - Worst case
  - So that’s 2n vs n^2 from before,
    - A big improvement!!
    - For example, if n = 100
    - n^2 = 10,000 steps, and 2n = 200 steps

Algorithm Implementation

- This problem took some creativity, but showed the basic elements of implementing an algorithm in C
  - In addition, you must know the proper syntax of C to program this.
  - There's no set way to implement an algorithm,
    - But this class will teach you ways to approach problems and analyze their complexity.

Algorithm Implementation

- Some of the questions we will ask when solving problems in this class will be:
  1) What data structures are going to be used?
  2) What functions are going to be needed?
  3) What exceptional cases or run-time errors should we check for?
  4) What is the most efficient way to execute the steps in the algorithm?

Assignment #1

- Due 9/7/2011, 2 weeks from this Wednesday.
  - Posted on the website later today.
- Mostly a review of things you have seen before:
  - structs, enumerations, reading from a file
- Things you haven't seen before
  - (we'll go over these things next class)
  - Dynamic memory allocation
  - Dynamically allocated array of structs